Programming Assignment 3: Data Visualisation and Dimensionality Reduction

Advances in Data Mining

Due Date: Oct 15, 23:59

Introduction

This assignment focuses on implementing the different approaches outlined in the lecture Data Visualisation and Dimensionality Reduction.

Task Descriptions

1. PCA on Lower-Dimensional Hyperplane

In this task, you will use a **make_blobs** dataset to demonstrate how PCA works when reducing a 3D dataset to 2D. You will first have to visualize the original 3D dataset. Then, you will standardize the data. After that, you will apply PCA. Finally, you will visualize the 2D projection and observe how the class structure appears in the lower-dimensional space.

2. PCA on Not Quite-Linear Data

In this task, you will work with a **3D dataset consisting of distinct Gaussian clusters**, one of which lie outside the primary hyperplane. This setup will demonstrate how PCA performs on data that is not entirely linear. You will begin by visualizing the original 3D dataset, along with its three orthographic projections: XY, XZ, and YZ, to better understand the positioning of the clusters. Next, you will standardize the data to ensure proper scaling. Afterward, you will apply PCA to reduce the dimensionality. Finally, you will visualize the 2D projection and observe how well PCA captures the underlying structure of the clusters.

3. PCA on Swiss-Roll-like Dataset

In this task, you will work with a **Swiss Roll** dataset, which has a complex, nonlinear structure unlike the simpler, more linear datasets from previous tasks. You will apply PCA to reduce the dimensionality, then visualize the original 3D data, the XZ projection, and the 2D PCA projection. Observe the results—did PCA capture the nonlinear structure well, or did it struggle?

4. t-SNE on Swiss-Roll-like Dataset

In this task, you will also work with the Swiss Roll dataset, but this time you will load it from the provided numpy files: `swiss_roll.npy` and `color.npy`. You will apply t-SNE for dimensionality reduction and plot the 2D projection. If you succeed in unfolding the roll using t-SNE, you will discover a letter (between A-Z) visible in the plot, and your last task is to print that letter.

5. Fine-Tuning t-SNE

In this task, we provide another dataset that contains a larger version of **the same letter** as in Task 4. Your task is to load this dataset (`swiss_roll_larger.npy` and `color_larger.npy`) and apply t-SNE again. This time, you should play with the parameters, consulting the slides to determine the best values, until the letter becomes visible in the plot and **larger** than the one produced in Task 4. Keep experimenting—if you think you can get an even bigger letter, keep adjusting the parameters to achieve the largest possible result.

Submission

You have been provided with five Python files, `task1.py` through `task5.py`, which contain the skeleton code for the tasks. You are also provided with the following numpy files: `swiss_roll.npy` and `color.npy` for Task 4, while `swiss_roll_larger.npy` and `color_larger.npy` for Task 5.

The code and comments should guide you on where to insert your own implementations. Please do **not** change any function names or return statements. Do not delete or modify any part of the skeleton code; instead, add your code in the designated sections.

Make sure all your code is well-commented and adheres to Python coding standards. You are only allowed to use the libraries already specified in the files.

Submission

For submission, you are required to rename all the skeleton files to `taskx_<xxxxxxx>.py`, where `<xxxxxxx>` is your student number without the leading 's'. For example, for student number s1005819 and task 1, the submission file should be named `task1_1005819.py`.

You are required to hand in the following **five** files:

- `task1_<xxxxxxx>.py`
- `task2_<xxxxxxx>.py`
- `task3_<xxxxxxx>.py`
- `task4_<xxxxxxx>.py`

- `task5_<xxxxxxx>.py`

Good luck with the assignment!